

Efficient Mining of Frequent Patterns from Uncertain Data

Carson K.-S. Leung, Chris Carmichael, Boyu Hao
Department of Computer Science
The University of Manitoba, Canada

ICDM-DUNE 2007

Outline

- Introduction & motivation
- Background & related work
- Proposed solution
 - **UF-tree**: A *tree structure* for capturing the content of transactions consisting of *uncertain data*
 - **UF-growth**: A *mining algorithm* for finding frequent patterns from the UF-tree
- Experimental results
- Conclusions

Leung et al. (U Manitoba, Canada)

Introduction

- **Data mining**
 - = non-trivial extraction of implicit, previously unknown, & potentially useful information from data
- Common data mining tasks for *pattern discovery*:
 - Clustering
 - Association rule mining / **frequent-pattern mining**
 - Etc.

Leung et al. (U Manitoba, Canada)

Motivation

- Since the introduction of frequent-pattern problem, lots of algorithms have been developed
 - E.g., Apriori-based algorithms, which rely on candidate generation & test
 - E.g., tree-based algorithms (e.g., FP-growth), which do NOT rely on candidate generation & test

Leung et al. (U Manitoba, Canada)

Motivation

- Many frequent-pattern mining algorithms were designed to handle *precise data*
 - Input: Transaction DB (e.g., market basket DB) where
 - Each transaction represents items that are purchased together by customers
 - or
 - Each transaction represents events that are co-occurring
 - Existence of each item (or event) in a transaction is *known*

Leung et al. (U Manitoba, Canada)

Motivation

- Some other real-life situations in which users deal with *uncertain data*:
 - Input: Transaction DB (e.g., patient diagnosis, biomedical records, sensor network data) where existence of each item in a transaction is best captured by a likelihood measure (say, *existence probability*)
- Want: Frequent-pattern mining algorithms for handling *uncertain data* (i.e., dealing with these situations)

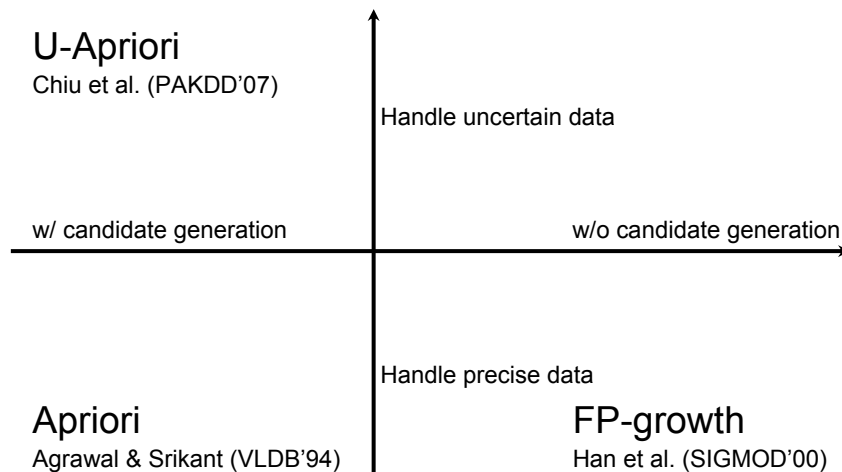
Leung et al. (U Manitoba, Canada)

Motivation

- Want: Frequent-pattern mining algorithms for handling *uncertain data*
- Response: *U-Apriori* algorithm
 - Apriori-based frequent-pattern mining algorithm for handling uncertain data
- When handling precise data, tree-based mining algorithms (e.g., FP-growth) are usually faster than Apriori-based ones
 - Q1: Would it be possible to handle uncertain data using *tree-based approaches*?
 - Q2: Would the resulting tree-based algorithm be *faster* than its Apriori-based counterpart (U-Apriori)?

Leung et al. (U Manitoba, Canada)

Motivation: Existing Algorithms



Leung et al. (U Manitoba, Canada)

Motivation: Our Proposal

U-Apriori

Chiu et al. (PAKDD'07)

UF-growth

Leung et al. (DUNE'07)

Handle uncertain data

w/ candidate generation

w/o candidate generation

Apriori

Agrawal & Srikant (VLDB'94)

Handle precise data

FP-growth

Han et al. (SIGMOD'00)

Leung et al. (U Manitoba, Canada)

Problem Statement

- We provide the user with a ***tree-based mining algorithm for finding frequent patterns from uncertain data***

Leung et al. (U Manitoba, Canada)

Contributions

- We propose:
 - **UF-tree**: A *tree structure* for capturing the content of transactions consisting of *uncertain data*
 - **UF-growth**: A *tree-based mining algorithm* for finding frequent patterns from the UF-tree

Leung et al. (U Manitoba, Canada)

Background

Frequent Patterns (for Precise Data)

- For precise data:
 - **Frequent patterns** are itemsets with **support** \geq user-defined minsup threshold
 - Since existence of each item in a transaction is known
 - **(Actual) support** of an itemset S in a transaction t_i is 1 if S is present in t_i
 - **(Actual) support** of an itemset S in a TDB can be obtained by counting #transactions that contain all the items in the itemset S

Leung et al. (U Manitoba, Canada)

Frequent Patterns (for Uncertain Data)

- For uncertain data:
 - **Frequent patterns** are itemsets with **support** \geq user-defined minsup threshold
 - Since existence of each item x in a transaction t_i is captured by its *existence probability* $P(x, t_i)$
 - **(Expected) support** of an itemset S in a transaction t_i is the expected probability (over all "possible worlds") of coexistence of all the items in S, i.e.,

$$\prod_{x \in S} P(x, t_i)$$

The value of expected support of S in t_i is (0, 1]
 - **(Expected) support** of an itemset S in a TDB can be obtained by summing over all transactions the expected probability of S, i.e.,

$$\sum_j [\prod_{x \in S} P(x, t_j)]$$

Leung et al. (U Manitoba, Canada)

Apriori vs. U-Apriori

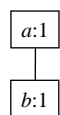
- Apriori (for handling *precise* data)
 - For each candidate itemset, compute the *actual* support
- U-Apriori (for handling *uncertain* data)
 - For each candidate itemset, compute the *expected* support

Leung et al. (U Manitoba, Canada)

Naïve Tree-Based Approach

- FP-growth / FP-tree (for handling *precise* data)
 - Each tree node captures its *actual* support in that tree path
- Naïve approach for handling *uncertain* data
 - Each tree node captures its *expected* support in that tree path

Tree



For precise data, TDB is:

TDB 0: $t_1 \{a, b\}$

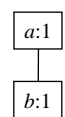
i.e., 1 transaction containing $\{a, b\}$

Leung et al. (U Manitoba, Canada)

Naïve Tree-Based Approach

- FP-growth / FP-tree (for handling precise data)
 - Each tree node captures its *actual* support in that tree path
- Naïve approach for handling uncertain data
 - Each tree node captures its *expected* support in that tree path

Tree



For uncertain data, TDB could be:

TDB 1: $t_1 \{a:1, b:1\}$

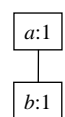
i.e., 1 transaction containing $\{a, b\}$ where
 $P(a, t_1)=1, P(b, t_1)=1$

Leung et al. (U Manitoba, Canada)

Naïve Tree-Based Approach

- FP-growth / FP-tree (for handling precise data)
 - Each tree node captures its *actual* support in that tree path
- Naïve approach for handling uncertain data
 - Each tree node captures its *expected* support in that tree path

Tree



For uncertain data, TDB could also be:

TDB 2: $t_1 \{a:0.5, b:0.5\}$
 $t_2 \{a:0.5, b:0.5\}$

Problem!

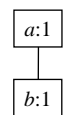
i.e., 2 transactions containing $\{a, b\}$ where
 $P(a, t_1)=0.5, P(b, t_1)=0.5; P(a, t_2)=0.5, P(b, t_2)=0.5$

Leung et al. (U Manitoba, Canada)

Naïve Tree-Based Approach

- FP-growth / FP-tree (for handling precise data)
 - Each tree node captures its *actual* support in that tree path
- Naïve approach for handling uncertain data
 - Each tree node captures its *expected* support in that tree path

Tree



For uncertain data, TDB could even be:

TDB 3: $t_1 \{a:0.1, b:0.2\}$
 $t_2 \{a:0.9, b:0.8\}$

Problem!

i.e., 2 transactions containing $\{a, b\}$ where

$$P(a, t_1)=0.1, P(b, t_1)= 0.2; P(a, t_2)= 0.9, P(b, t_2)= 0.8$$

Leung et al. (U Manitoba, Canada)

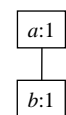
Naïve Tree-Based Approach

- FP-growth / FP-tree (for handling precise data)
 - Each tree node captures its *actual* support in that tree path
- Naïve approach for handling uncertain data
 - Each tree node captures its *expected* support in that tree path

TDB 0: $t_1 \{a, b\}$

From TDB 0 (for precise data),
actual support of $\{a, b\}$ is 1

Tree



From the tree, we also get
actual support of $\{a, b\}$ is 1

Leung et al. (U Manitoba, Canada)

Naïve Tree-Based Approach

- FP-growth / FP-tree (for handling precise data)
 - Each tree node captures its *actual* support in that tree path
- Naïve approach for handling uncertain data
 - Each tree node captures its *expected* support in that tree path

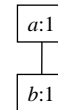
TDB 2: $t_1 \{a:0.5, b:0.5\}$
 $t_2 \{a:0.5, b:0.5\}$

From TDB 2,
expected support of $\{a, b\}$ is:
 $0.5*0.5 + 0.5*0.5$
 $= 0.25 + 0.25 = 0.50$

TDB 3: $t_1 \{a:0.1, b:0.2\}$
 $t_2 \{a:0.9, b:0.8\}$

From TDB 3,
expected support of $\{a, b\}$ is:
 $0.1*0.2 + 0.9*0.8$
 $= 0.02 + 0.72 = 0.74$

Tree



Problem! *How to get expected support of $\{a, b\}$ from the tree?*

Leung et al. (U Manitoba, Canada)

Our Proposed Tree Structure: UF-tree

UF-tree

- Instead of just storing item & expected support, each node in our UF-tree stores:
 - Item
 - Expected support
 - *Occurrence* (i.e., number of transactions containing such an item)

TDB 1: $t_1 \{a:1, b:1\}$

UF-tree From TDB 1, expected support of $\{a, b\}$ is:
 $1 * (1*1) = 1$

(a:1):1

(b:1):1

We can get the same info from our UF-tree!

TDB 2: $t_1 \{a:0.5, b:0.5\}$
 $t_2 \{a:0.5, b:0.5\}$

UF-tree From TDB 2, expected support of $\{a, b\}$ is:
 $2 * (0.5*0.5) = 0.50$

(a:0.5):2

(b:0.5):2

Again, we can get the same info from our UF-tree!

Leung et al. (U Manitoba, Canada)

Our Proposed Algorithm: UF-growth

UF-growth

1. Build an UF-tree to capture uncertain data
2. Find frequent patterns from the UF-tree

Leung et al. (U Manitoba, Canada)

UF-growth

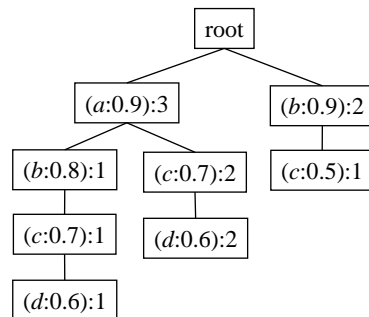
1. Build an UF-tree to capture uncertain data

TDB:

t_1	{ $a:0.9, b:0.8, c:0.7, d:0.6, f:0.8$ }
t_2	{ $a:0.9, c:0.7, d:0.6, f:0.1$ }
t_3	{ $b:0.9, c:0.5, e:0.4$ }
t_4	{ $b:0.9, e:0.2$ }
t_5	{ $a:0.9, c:0.7, d:0.6, e:0.3$ }

UF-tree for TDB: minsup=1.0

Header Table	
<i>Item</i>	<i>expSup</i>
a	2.7
b	2.6
c	2.6
d	1.8

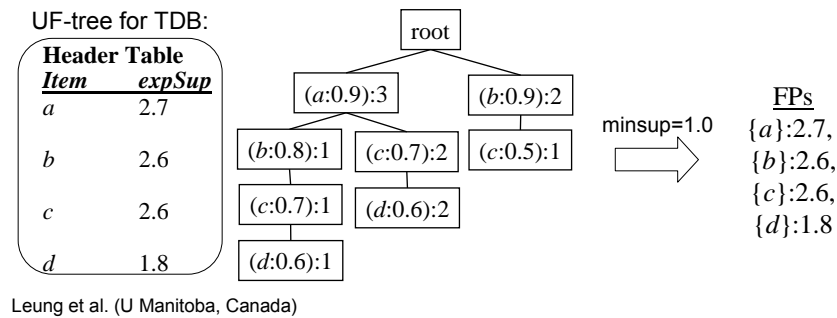


Leung et al. (U Manitoba, Canada)

UF-growth

2. Find frequent patterns from the UF-tree

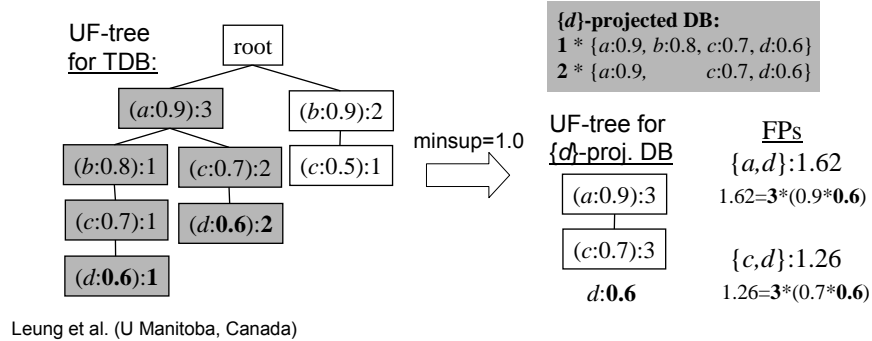
- **Challenge:** Expected support of an itemset S is computed by $\sum_i [\prod_{x \in S} P(x, t_i)]$, i.e., sum of product of $P(x, t_i)$'s



UF-growth

2. Find frequent patterns from the UF-tree

- **Challenge:** Expected support of an itemset S is computed by $\sum_i [\prod_{x \in S} P(x, t_i)]$, i.e., sum of product of $P(x, t_i)$'s

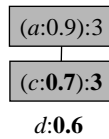


UF-growth

2. Find frequent patterns from the UF-tree

- Challenge: Expected support of an itemset S is computed by $\sum_i [\prod_{x \in S} P(x, t_i)]$, i.e., sum of product of $P(x, t_i)$'s

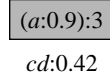
UF-tree for $\{d\}$ -projected DB



minsup=1.0

$\{c,d\}$ -projected DB:
3 * {a:0.9, cd:0.6*0.7}

UF-tree for $\{c,d\}$ -proj. DB



FP

{a,c,d}:1.13
1.13=3*(0.9*0.42)

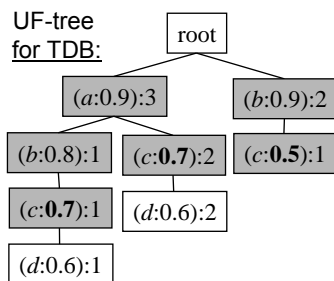
Leung et al. (U Manitoba, Canada)

UF-growth

2. Find frequent patterns from the UF-tree

- Challenge: Expected support of an itemset S is computed by $\sum_i [\prod_{x \in S} P(x, t_i)]$, i.e., sum of product of $P(x, t_i)$'s

UF-tree for TDB:

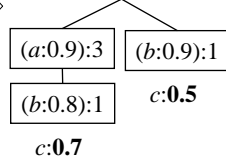


minsup=1.0

$\{c\}$ -projected DB:

1 * {a:0.9, b:0.8, c:0.7}
2 * {a:0.9, c:0.7}
1 * { b:0.9, c:0.5}

UF-tree for $\{c\}$ -proj. DB



FPs

{a,c}:1.89
1.89=3*(0.9*0.7)

{b,c}:1.01
1.01=1*(0.8*0.7)+
1*(0.9*0.5)

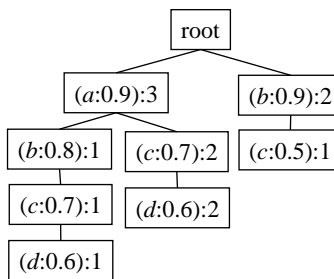
Leung et al. (U Manitoba, Canada)

UF-growth

1. Build an UF-tree to capture uncertain data
2. Find frequent patterns (FPs) from the UF-tree

UF-tree for TDB:

Header Table	
<i>Item</i>	<i>expSup</i>
<i>a</i>	2.7
<i>b</i>	2.6
<i>c</i>	2.6
<i>d</i>	1.8



minsup=1.0

FPs
 {*a*}:2.7,
 {*b*}:2.6,
 {*c*}:2.6,
 {*d*}:1.8,
 {*a,d*}:1.62,
 {*c,d*}:1.26,
 {*a,c,d*}:1.13,
 {*a,c*}:1.89,
 {*b,c*}:1.01

Leung et al. (U Manitoba, Canada)

Experimental Results

Experimental Results

- Experiments on typical test database for data mining (e.g., IBM synthetic data, UCI real data)
- Evaluate the effectiveness of our proposed UF-growth
- Key observations:
 - Runtime of UF-growth < runtime of U-Apriori
 - minsup \uparrow \rightarrow #frequent patterns \downarrow \rightarrow runtime \downarrow
 - Linear scalability

Leung et al. (U Manitoba, Canada)

Conclusions

Conclusions

- We provided the user with a ***tree-based mining algorithm for finding frequent patterns from uncertain data***
- Specifically, our proposal consists of:
 - ***UF-tree***: A *tree structure* for capturing the content of transactions consisting of *uncertain data*
 - ***UF-growth***: A *tree-based mining algorithm* for finding frequent patterns from the UF-tree

Leung et al. (U Manitoba, Canada)

Conclusions

- Q1: Would it be possible to handle uncertain data using *tree-based approaches*?
 - Answer: Yes, using our ***UF-growth algorithm*** with the *UF-tree*
- Q2: Would the resulting tree-based algorithm be *faster* than its Apriori-based counterpart (U-Apriori)?
 - Answer: Yes, experimental results showed runtime of the resulting tree-based *UF-growth* algorithm < runtime of *U-Apriori*
- The UF-growth algorithm effectively uses UF-tree to capture uncertain data & efficiently finds frequent patterns from these uncertain data

Leung et al. (U Manitoba, Canada)

Thank you / Merci

kleung@cs.umanitoba.ca
www.cs.umanitoba.ca/~kleung